Analysis of Bitcoin Network Dataset for Fraud

Deepak Zambre, Ajey Shah
Stanford CS 224W Project Final Report - Group 30
deepak.zambre@gmail.com, ajey.shah@gmail.com

December 10, 2013

Abstract

Bitcoin uses peer-to-peer technology to operate with no central authority or banks; managing transactions and the issuing of bitcoins is carried out collectively by the network. Bitcoin is open-source; its design is public, nobody owns or controls Bitcoin and everyone can take part [1]. Time and again Bitcoin users have reported their Bitcoin wallets being compromised [2]. Ready availability of all transaction information in Bitcoin network makes it easy to trace the flow of money from compromised accounts in the wake of reported heist or robbery. This project aims to identify peculiar properties of users carrying out such heists which make them stand out.

1 Introduction

In Bitcoin network, all transactions take place between public keys owned by Bitcoin users. Each user can own more than one public key, which are grouped together in the users wallet. The public key ownership data for users is not publicly available although all the transactions in Bitcoin network are available to be seen and analyzed on [3]. This allows Bitcoin network to have unprecedented levels of transparency. Once a public key is involved in a transaction that public key is tainted with history with of all transactions of all the public keys that are involved in the new transaction.

Figure 1 shows a sample Bitcoin transaction with one input and one output [4]. The input for this transaction is from the output of another transaction whose key is contained in the field 'Previous tx'. Because all transactions are public the network knows that this prior transaction resulted in an output of 50 BTC. The output for this transaction is the user represented by the field "scriptPubKey'. It also contains the amount of bitcoins that were transferred, in this case 50 BTC. In interest of brevity, we only explain the parts relevant to our project. The remaining details are found at [4].

Figure 2 shows a more complex transaction [4]. In this case transaction C has two inputs, one each from the outputs of transaction A and transaction B. In all, the inputs represent 150 BTC. But C wants to only transfer 101 BTC to

user in transaction D. So the remaining amount in transaction C, i.e. 49 BTC, is transferred to C again as a balance. In transaction D, we see that use has sent 101 BTC he received from transaction C to someone else, but that person hasn't yet claimed their share. More in depth discussion is out of scope for this paper and the authors recommend a thorough reading of [4].

```
Input:
Previous tx: f5d8ee39a430901c91a5917b9f2dc19d6d1a0e9cea205b009ca73dd04470b9a6
Index: 0
scriptSig: 304502206e21798a42fae0e854281abd38bacd1aeed3ee3738d9e1446618c4571d10
90db022100e2ac980643b0b82c0e88ffdfec6b64e3e6ba35e7ba5fdd7d5d6cc8d25c6b241501

Output:
Value: 5000000000
scriptPubKey: OP_DUP OP_HASH160 404371705fa9bd789a2fcd52d2c580b65d35549d
OP_EQUALVERIFY OP_CHECKSIG
```

Figure 1: Sample transaction in Bitcoin Network [4]

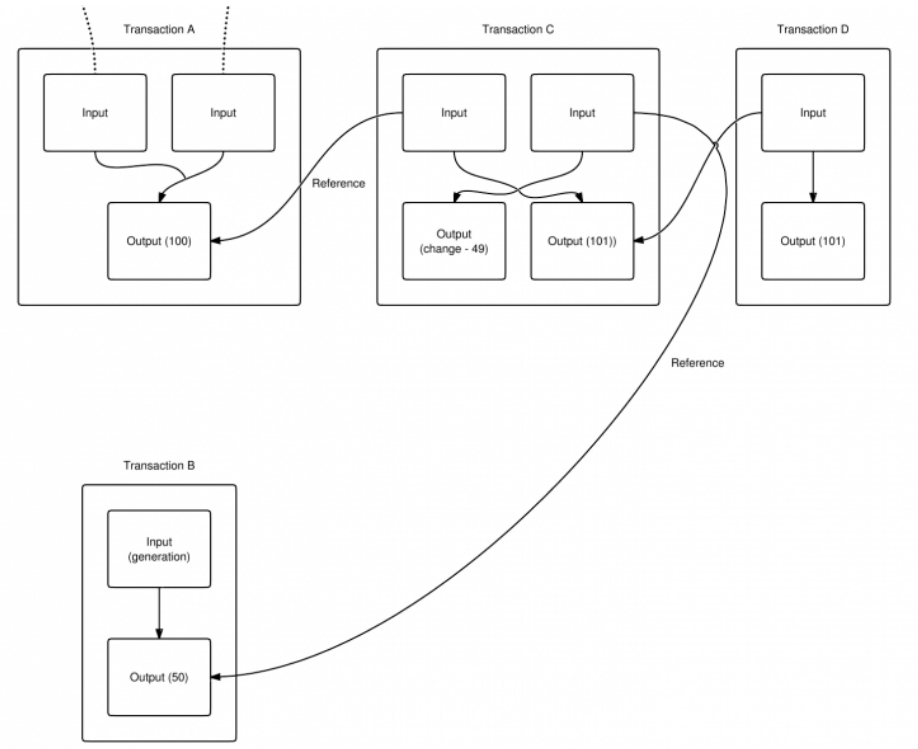


Figure 2: Transaction traceability in Bitcoin network [4]

Traceability of all transactions made by public keys shifts the onus of maintaining anonymity to the users of the public keys. Although Bitcoin mentions

some preventive steps to safeguard one's identity and public keys, Bitcoin theft cannot be prevented totally. [2] lists some of such know reported Bitcoin heists. In this project we are going to analyze the Bitcoin blockchain transaction information particularly for All In Vain Theft, Stone Man Loss and Mass Bitcoin Thefts as reported in [2]. We will attempt to unravel a series of user features with focus on clustering users. The aim of clustering will be to separate out users exhibiting behavior similar to the users who carried out the aforementioned thefts.

We begin with section 2 where we look at previous attempt to detect fraud in Bitcoin Network, online auction systems and file reputation. We briefly describe the pattern of aforementioned robberies in section 3. In section 4, we describe the Bitcoin dataset that we have used, we briefly describe basic properties of this graph and then move on to explain the attributes that we mine from this network data on which we fit the kMeans algorithm. In section 5 we show how our model performs in trying to cluster the robbers away from good user clusters, we also show evaluation of the model on generated test data. Thereafter, we provide insights on why the model works by analyzing some interesting properties of robbers in section 6. Finally, we conclude in section 7.

Domain difficulties and non-goals

We depend exclusively on the reported robberies for our feature design exercise. This has been true for [5] which performed anonymity analysis for Bitcoin network. This makes us dependent on assumed veracity of the reporting done in [2]. Since it is out of scope of the project to corroborate this, we assume that [2] is a true account of reported robberies.

Although all the Bitcoin transaction data is openly available, paradoxically, we would like to mention that data available in connection to heists is small. This becomes extremely important when we restrict ourselves to analyzing the Bitcoin dataset before July 2011. Presence of mixing services without enough information to filter out operations performed by these services further complicates the matters because the behavior exhibited by mixing services might have representative features of robberies.

2 Prior art

Anonymity of Bitcoin network and its repercussions have been analyzed in [5]. In [5] authors perform passive analysis on Bitcoin network data to claim that Bitcoin system is far from being anonymous. Reuse of public keys in some Bitcoin clients, ip-address logging for transactions, centralized services like wallet services, Bitcoin exchanges having ready access to user information, among several others are enough to trace the owner of public keys, thus thwarting the anonymity in Bitcoin network. [5] also perform temporal analysis of All in Vain Theft [2] on the Bitcoin network. The flow of Bitcoin during this theft is analyzed by laying out the public key network used by the thief.

[6] describes system design for fraud detection in online auction networks. Netprobe models auction data as a Markov Random Field (MRF), wherein each entry represents the probability of a node being in a particular state given that its neighbors are in respective particular states. Thereafter, Belief Propagation algorithm [8] is run on the represented auction data model to infer maximum likelihood state probabilities of nodes in MRF. The paper claims that the aforementioned process allows them to assign rank to users, in the same spirit as PageRank assigns rank to web pages. Using [9] the paper claims that detection of fraud in auction networks can be reduced to detection of near bipartite cores for which they introduce scalable Increment NetProbe which does belief propagation on h-vicinity of a node n.

In [7] the authors present an algorithm based off of belief network propagation algorithm to infer a files reputation and flagging files with low reputation as malware. They ran the tests on a large graph consisting of 1 billion nodes and 37 billion edges. Nodes consisted of machines and files in a bipartite network. They show, based off of empirical data from their tests, that the Polonium algorithm is effective in improving accuracy in detecting malware.

3 Robbery Description

3.1 Stone Man Loss(SML)

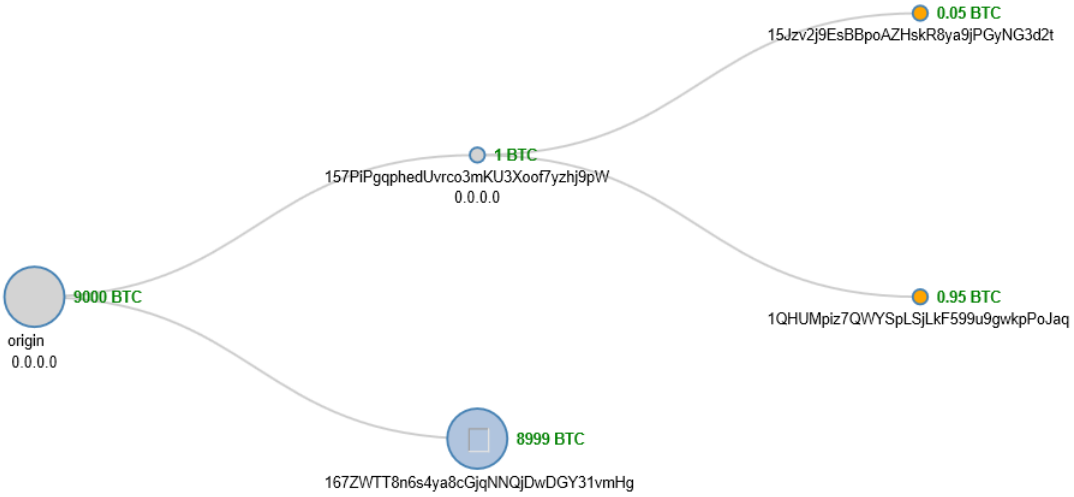
The transaction of interest *eb5b761c7380ed4c6adf688f9e5ab94953dcabeda47d9eeabd77261902fccccf* is shown in figure 3a. The transaction took place on August 09, 2010, 11:35:00 PM wherein 8999 Bitcoins were robbed from the original key. This was the single transaction that was associated with this robbery.

3.2 All In Vain(AIV)

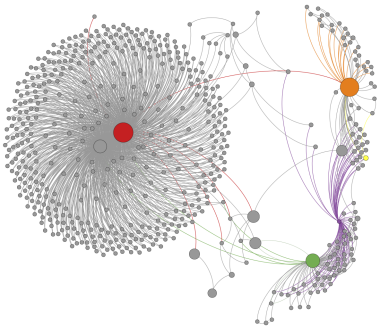
AIV robbery is shown in figure 3b. Green node shows the victim and red node shows the robber. Actual theft of 25,000 Bitcoins was preceded by theft of 1 Bitcoin. The rogue user manufactured a large number of transactions after the theft in order to taint the bitcoins. The network induced due to these transactions is shown in figure 3b(for simplicity, the network shown in limited to show level 2 nodes from rouge user). More details on theft can be found by referring [2] and [5].

3.3 Mass MyBitcoin Theft(MMT)

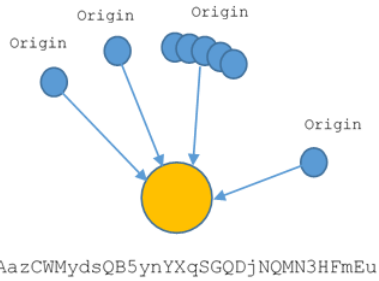
This theft occured in June, 2011 when users having same password on MyBitcoin and Mt.Gox were compromised because their Mt.Gox passwords were leaked. The thief transferred bitcoins from all the compromised keys to his key, see figure 3c. The theft involves a large number of transactions, one for each compromised user, where each transaction steals bitcoins from the compromised user to the rogue user.



(a) Stone man loss robbery[2]



(b) All in vain robbery[5]



(c) MyBitcoin theft[2]

Figure 3: Robberies

Classifying the rogue users in SML, AIV and MMT separate from the respective victims is within the scope of this project.

4 Attribute extraction and model description

4.1 Dataset Description

We have used publicly available data from [10]. This dataset consists of transaction information of the Bitcoin Network until July 13, 2011. Table 1 shows brief statistics of the dataset that we have used. The dataset consists of two files:

Total number of nodes	881,678
Total number of edges	1,617,212
Victims aka good users	628
Robbers aka rogue users	3

Table 1: Dataset Statistics

4.1.1 User Vertices File

This file is used to identify public keys used by users in Bitcoin network. Each line in this file consists of a tab-separated public keys belonging to a single user in the Bitcoin Network. The line number in this file is used as identifier for the user.

4.1.2 User Edges File

This file consists of transaction data for Bitcoin Network. Each line in this file represents transaction occurring in the Bitcoin Network. In each line, the first column represents the user id of the user who gives bitcoins, the second column represents the user who receives bitcoins, the third column represents the amount of bitcoins transferred and the fourth column represents the date-time when this transaction took place.

4.1.3 Robbery Data Collection

We identified the keys involved in all the aforementioned robberies by crawling *www.blockchain.info* and cross-checking the transaction data that we crawl with the data that we already have. Thereafter, we searched the user ids of the respective keys to gather ids of 3 rogue users (one for each previously described robbery) and 628 victims (representing all the users that were robbed).

4.2 Feature extraction

Our aim was to come up with features that will segregate the rogue users from good users. In this section, we define the features that we experimented with for our model (not all features in this section gave the optimal results, further details are mentioned in section 5), specifically, the features with their names underlined performed optimally for us.

1. *User in degree*: represents the number of incoming transactions to a user
2. *User out degree*: represents the number of outgoing transactions from a user
3. *User unique in degree*: represents the number of unique users from which the user has received money

4. *User unique out degree*: represents the number of unique users to which a user sent money to

5. *User mean in*: represents the mean value of bitcoins received by a user

6. *User mean out*: represents the mean value of bitcoins sent by a user

7. *User in standard deviation*: represents the standard deviation of bitcoins received by the user

8. *User out standard deviation*: represents the standard deviation of bitcoins sent by a user

9. *Number of in-transactions per unique in-transaction timestamp*: represents the ratio of the number of incoming transactions to the number of unique timestamps for those transactions

10. *Number of out-transactions per unique out-transaction timestamp*: represents the ratio of the number of outgoing transactions to the number of unique timestamps for those transactions

11. *In-transaction rate*: represents the outgoing transaction frequency for the user

12. *Out-transaction rate*: represents the outgoing transaction frequency for the user

13. *Balance*: represents the amount of bitcoins owned by the user

14. *Total known user keys*: represents the total number of keys used by the user

15. *Average in-velocity*: represents the speed with which bitcoins flow to a user

16. *Average out-velocity*: represents the speed with which bitcoins flow from a user

17. *In-velocity standard deviation*: represents the standard deviation of the in-velocity for a user

18. *Out-velocity standard deviation*: represents the standard deviation of the out-velocity for a user

19. *Average in-acceleration*: represents the acceleration of bitcoin flow to a user

20. *Average out-acceleration*: represents the acceleration of bitcoin flow from a user

21. *Average neighbor out velocity*: represents the average rate of bitcoin flow out of a user's neighbor nodes

Figure 4 shows log-log frequency plots for some of the aforementioned attributes. User in/out degree, user balance, user know keys exhibit graph similar to power law. This is indicative of the fact that there a few users that transact on a very high scale. These people also use a large number of public keys, since Bitcoin recommends using a new public key per transaction to maintain anonymity. The graph does not exhibit high clustering coefficient, indicative of the fact that the transaction data is not similar to friendship network in this aspect, neither does the graph demonstrate high index for triadic closure.

It is worthwhile to note that the plots shown in figure 4 for velocity, acceleration of user and neighbor velocity also resemble power law. The rogue users identified by red dots can clearly be seen to be occurring with low frequency. This is indicative of the fact that these attributes can be used to identify the rogue users. In 5 we will see that indeed, these properties are a part of collection of attributes that give optimal results for us.

4.3 Model description

We extracted the aforementioned features for each user in the dataset and then used kMeans clustering to group users together. The main idea of kMeans clustering is to define K centroids and assign each point in the dataset to one of these k centroids so that the following objective function is minimized,

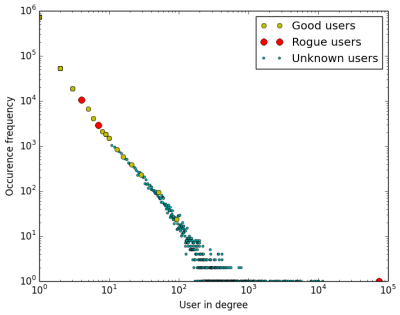
$$\sum_{j=1}^{k}\sum_{i=1}^{n}||x_i^{(j)} - c_j||^2,$$

where datapoint $x_i^{(j)}$ is assigned to cluster c_j. Further details of kMeans can be found in [11].

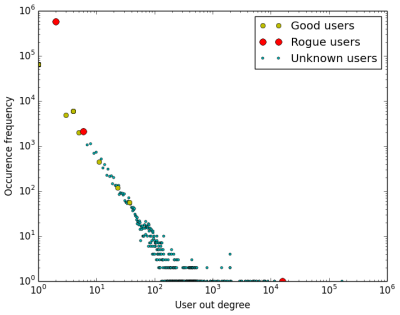
We ran kMeans on dataset generated using *(a) all aforementioned features* and *(b) subset of aforementioned features (subset was determined by seeing if the results of clustering separated good users from rogue users).*

Feature Scaling and mean normalization

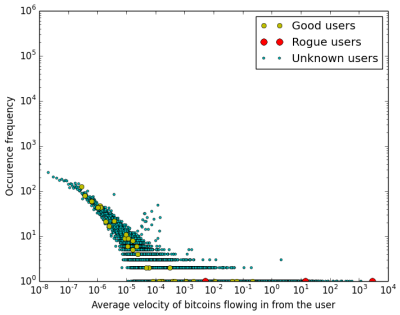
KMeans is susceptible to feature bias if one of the feature is having significantly larger range than some other feature. To avoid this, we used feature scaling and normalization on the extracted features. However, in our case, using feature scaling and normalization didn't yield results significantly different from the results that we got without using feature scaling and normalization.



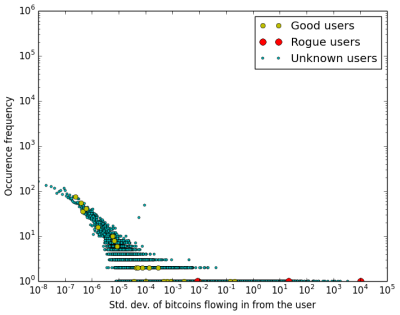
(a)



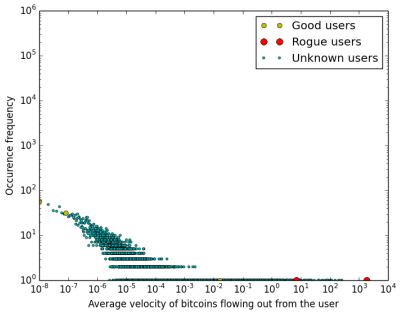
(b)



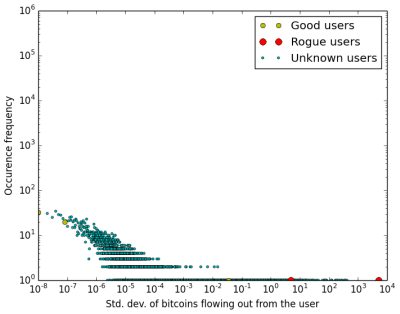
(c)



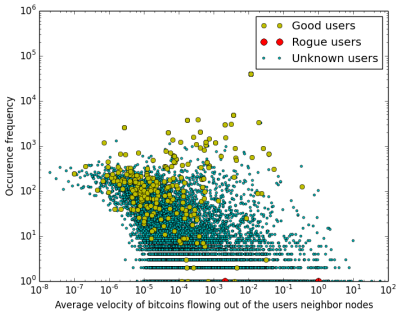
(d)



(e)



(f)



(g)

Figure 4: Various attribute plots for the dataset

5 Evaluation

5.1 kMeans on All Features

We got underwhelming results when we tried to cluster users represented by the previously features. The generated kMeans model failed to classify the robber nodes in a different cluster from the good users. The resultant model was one huge cluster with 99% nodes in it.

5.2 kMeans on Selective Features

After testing many different feature combinations we settled on feature combination mentioned in table 2 as best representative model for analyzing robberies in the bitcoin network.

Feature	Description
In-transaction rate	outgoing transaction frequency for the user
Average in-velocity	speed with which bitcoins flow to a user
Average out-velocity	speed with which bitcoins flow from a user
Average in-acceleration	acceleration of bitcoin flow to a user
Average out-acceleration	acceleration of bitcoin flow from a user
Average neighbor out velocity	average rate of bitcoin flow out of a user's neighbor nodes

Table 2: Features giving optimal classification results of rogue users for kMeans algorithm

With users represented using features from table 2 we were able to classify the rogue users in a cluster different from the good user's cluster. Figure 5 shows the plot for kMeans cost function against the number of clusters. We select $k = 5$. **We observed two significant clusters in the resultant kMeans model. First cluster with *756916 nodes* had all the 628 good user nodes. Another cluster with *124761 nodes* had the 3 rogue user nodes. Thus we were able to classify the good users separately from the rogue users.** We hypothesize that the other nodes in the rogue user's cluster may represent the mixing services that are prevalent in Bitcoin Network, however, we do not have any reported data on the mixing services whatsoever.

5.3 Synthetic tests

We generated node data resembling robbery patterns as mentioned in section 3. We used "*robber distance length*" (signifies the shortest distance between the thief and the victim) and "*expansion*" (represents the expansion induced due the manufactured transactions by thief) as the parameters for generating the synthetic node data for thief. We assumed that the thief robbed the victim of all his bitcoins in this network generation process. Using this approach of synthetic node data generation we were able to create robberies resembling the

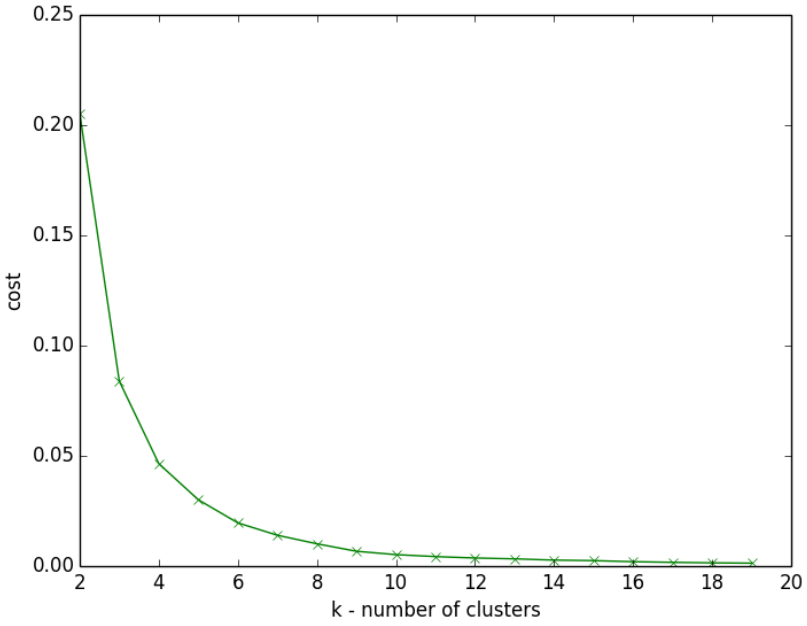


Figure 5: kMeans cost function against the number of clusters

three robberies that we focused on. **We varied both the parameters from 1 to 10 (thus giving 100 rogue users) and were able to achieve** 76.5% **precision rate in clustering the test rogue users in the cluster same as the actual rogue user's cluster**.

6 Insights on model

As we can see from section 5, kMeans performed well when we used a subset of all the attributes that we had extracted. We confirm that these properties are indeed indicative of the thefts that had occurred. Lets define new attributes as,

$$indicator_1 = \frac{1}{1 + e^{-1 * \mu(in\_velocity)^{\sigma(in\_velocity)}}}$$

$$indicator_2 = \frac{1}{1 + e^{-1 * \mu(in\_bitcoins)^{\sigma(in\_bitcoins)}}}$$

$$indicator_3 = \frac{1}{1 + e^{-1 * \mu(in\_transaction\_rate)^{\sigma(in\_transaction\_rate)}}}$$

where,

$$\mu(X) = mean\ of\ attribute\ X,$$

$$\sigma(X) = standard\ deviation\ of\ attribute\ X$$

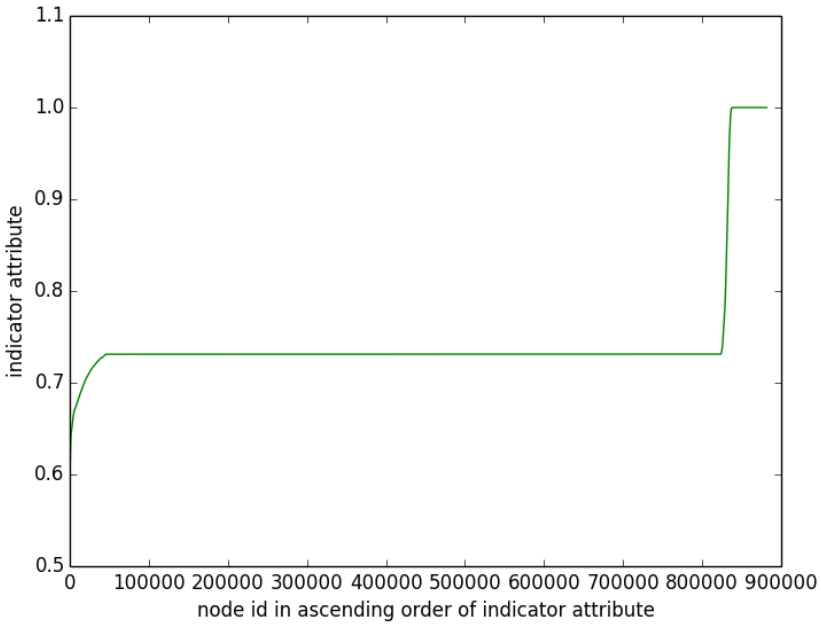


Figure 6: Plot for indicator attribute

Figure 6 shows the plot of $indicator_3$ against *node id* (where smaller node id indicates smaller indicator attribute value). Similar plots were obtained for other indicator attributes. The rogue users [section 3] appear on the rightmost side of the green line (where as the good users are mostly in the left / middle region of the plot). This is indicative of observed wide swings of attribute values for attributes such as in_velocity, in_bitcoins and in_transaction_rate for rogue users. Practically, it also means that the keys used for thefts are not regularly used for thefts and hence it becomes easier to spot the swing in the transactional behavior of the user. However, one shortcoming of this model is that it will not be very successful for robberies that happen over an extended period of time in small increments. However, we see that the bulk of the reported robberies in Bitcoin Network are not of this nature.

7 Conclusion

Our main contribution to the problem of identifying rogue users is a model comprised of 6 unique features as seen in table 2. While not a hundred percent

accurate, we feel that this model is a good starting point to analyze bitcoin users to find potential rogue users. In our tests based on real reported robberies we were able to separate good users from the rogue users. For synthetically generated rogue users, we were able to achieve 76.5 percent accuracy.

References

[1] `http://bitcoin.org/en`

[2] `https://bitcointalk.org/index.php?topic=83794.0#post_ubitex_scam`

[3] `http://blockchain.info/`

[4] `https://en.bitcoin.it/wiki/Transactions`

[5] Reid, F. and Harrigan, M. *An Analysis of Anonymity in the Bitcoin System* 2011.

[6] Shashank Pandit, Duen Horng Chau, Samuel Wang, Christos Faloutsos *NetProbe : A Fast and Scalable System for Fraud Detection in Online Auction Networks* 2007: Data Tracking

[7] Duen Horng (Polo) Chau, Carey Nachenberg, Jeffrey Wilhelm, Adam Wright, Christos Faloutsos *Polonium: Tera-Scale Graph Mining and Inference for Malware Detection* 2011, Proccedings of SIAM International Conference on Data Mining (SDM)

[8] xyz *Understanding Belief Propagation and its generalizations*

[9] xyz *Detecting Fraudulent personalities in networks of online auctioneers*

[10] `http://anonymity-in-bitcoin.blogspot.com/2011/09/code-datasets-and-spsn11.html`

[11] J. B. MacQueen *Some Methods for classification and Analysis of Multivariate Observations* 1967, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability